Lessons Learned from the Failure of an Experience Base Initiative Using a Bottom-up Development Paradigm

Arne Koennecker, University of Kaiserslautern/Fraunhofer IESE
Ross Jeffery, University of New South Wales
Graham Low, University of New South Wales

Address for correspondence:

Ross Jeffery
Centre for Advance Empirical Software Research
School of Information Systems
University of New South Wales
Sydney 2052 NSW
Australia
Email: r.jeffery@unsw.edu.au

Abstract.
This paper describes the development of an experience factory in an Australian organization. Information structures were well developed and used in the daily work of the organization. This included the use of network technology as well as the personal interaction between department members. Highly motivated personnel drove improvement via new techniques, knowledge, and tools. A special focus existed to simplify work tasks through tool support. Daily work and problem solving was strongly based on personnel interaction and access to knowledge bases (documentation, mail lists, etc.). The goal of the project was to package personnel experience and best practices and provide an effective framework for access and integration. The system was decommissioned shortly after the completion of the project. The reasons for this are discussed.

1. Introduction

Faced with improvement needs, in 1998 the company started to put special attention on approaches to support improvement activities in a structured way. Like many organizations in the software industry, improvement aspects and strategy issues ranged from product quality and project management to the overall improvement of software engineering skills. Further local improvement aspects had been identified in software process assessment using the CMM (Capability Maturity Model from the SEI [1]) and the ISO 9001 standard.

At the end of 1998, a project was started in co-operation with The Centre for Advanced Empirical Software Research (CAESAR) to evaluate the Experience Factory (EF) / Quality Improvement Paradigm (QIP) [2] concept. The concept was to be evaluated as an approach to support local improvement activities and to be applied as an approach in the given environment at the R&D department. The aim was to find a suitable approach within six months and to start realization of benefits as early as possible.

The choice of the EF / QIP concept was motivated by several aspects. Firstly, it was seen to be a promising concept that had been the subject of research projects in the past such as PERFECT (ESPRIT III project, sponsored by the CEC [3]). Secondly, the concept had already been applied in other organizations such as the Software Engineering Laboratory at NASA [4] and Daimler Benz AG [5]. Thirdly, the EF/QIP concept reflects the state-of-the-art in the field of improvement approaches, and therefore is of interest to the company.

The focus of the project was guided by five questions.

(1) Where has the EF concept already been applied, and what have been the experiences with it?
(2) What are the important characteristics of the company's environment, and of the company's philosophy, which need special consideration?
(3) Is the EF approach applicable considering the environment specifics in the organization.
(4) If (3) is true: How has the EF approach to be tailored so that it fits the needs and characteristics of the organization?
(5) If (3) is false: How can an organization-specific approach be developed which considers EF principles?

Principles of the classical EF approach

The EF approach describes an organizational framework, which addresses the issues of product and process improvement in software development organizations by providing an environment for continuous improvement. The EF approach defines an environment for controlled experimentation, knowledge reuse, experience packaging, and analysis of the development processes. The improvement environment consists of two parts: the project organization (PO) and the experience factory organization (EFO). Each of these follows

distinct steps in the Quality Improvement Paradigm (QIP). The project organization's major aim is to deliver software products according to given requirements. The PO uses information to improve, say, the product quality, the project performance or the reliability of project planning.

The Quality Improvement Paradigm

The QIP is the main driving force for continuous improvement and is integrated in both the PO and the EFO. It is defined as consisting of six steps [6]:

1. *Characterize* the current project and its environment with respect to existing models and metrics.
2. *Set the quantifiable goals* for successful project performance and improvement based on the first step and the business and project specific goals.
3. *Choose the appropriate process* model and supporting methods and tools for the project and define a project plan, which considers the decisions and definitions made in steps 1 and 2.
4. *Execute the process*, construct the products, collect and validate the data, and analyze it to provide real time feedback.
5. *Analyze the data* and evaluate the current practices, determine problems, record findings, and make recommendations for future project improvements.
6. *Package the experience* in the form of updated and refined models and other forms of structured knowledge gained from this project. Save it in an *Experience Base* to be reused in future projects.

The PO interacts during the project with the EF organization (EFO). The EFO supports it with knowledge and experience gained in the past and provides feedback about the performance and quality of the current project while analyzing the data provided. The task of the EFO, besides support during the software life cycle, is to package experience gained during projects in a reusable form and to store it in an Experience Base (EB).

The interacting PO and EFO realize two feedback loops, a project feedback loop that takes place in the execution phase (support & analysis), and an organizational feedback loop that takes place after a project is completed (analysis & packaging). The second feedback loop changes or improves the organization's understanding of software development by packaging and reusing experience and making it accessible to future projects.

How to build and run an EF

To start an EF there are two possible approaches: a top-down or a bottom-up approach. That is proceeding from defining processes, structures, products, and responsibilities to collecting concrete experience data, or else collecting data and proceeding back up a similar hierarchy. Basili and McGarry [7] propose a top-down approach, which aims to define and establish the required elements before the improvement activities and the data collection takes place. This provides a guiding, and more or less stable structure and the time to focus on analysis of results and products rather than on integrating changes in the structure while working with them. Five key steps characterize the described top-down approach: (1) Obtain commitment, (2) Establish structure (3) Establish processes (4) Produce baseline (5) Identify potential changes.

The EF at the SEL-NASA

The Software Engineering Laboratory (SEL) was started in 1976 at the NASA / GSFC comprising three organizations: NASA / GSFC Flight Dynamics Division, University of Maryland (Department of Computer Science), and the Computer Science Corporation (Flight Dynamics Technology Group). Its goal was to understand and improve the software development process and products within the GSFC Flight Dynamics Division. In this environment the EF concept was developed and first published in 1985 by V. Basili (with a later version in [2]) as a concept based on the research and experience of the SEL. Since then the EF has been successfully applied in the NASA environment and used in more than one hundred projects dealing with different improvement issues and technologies. The experiences range from detected impacts through the use of EF on product and process attributes, to recommendations as to what to consider when establishing an EF.

The EF at the Daimler Benz AG

Software plays a major role in the product range at Daimler-Benz. Outside of the SEL, the Daimler Benz experience is the only other report directly related to the establishment of an EF in a practical development environment. Furthermore, they describe their experiences in the first year of the EF project, which was significant to our need to establish benefits in a short time period. Three separate projects formed the basis of analysis. Project A was in the aerospace domain with mainly in-house software development of large embedded systems and rigid real-time constraints. A measurement program had already been commenced. The goals were to make improvement efforts persistent and repeatable, project effort predictable, and to support technical reviews. The

initiative comprised two application projects and 2-3 people were concerned with EF activities. Project B involved small-embedded systems. The development changed from contractors to in-house in recent years. The goal was to build core competencies and clarify development questions such as how to keep software portable, and how to make sure that each planned function was implemented. Review techniques were identified as potential support for this. The initiative comprised 1-2 application projects and 2-8 people were concerned with EF activities. Project C dealt with large administrative software units for managing internal business processes. Software requirements were defined in-house, but the development was outsourced. The focus for the EF was quality assurance, especially in outsourced development. In this case the initiative comprised 3 application projects and 2-3 people were concerned with EF activities.

For projects A and B the company followed the top-down approach discussed above. The measurement of the baseline started several months after the EF initiative. This first stage consisted of the definition of essential EF structures, processes, roles, and products. They also decided in project B to assist technical reviews and collect related data to help solve current problems. This was done without defining structures, and is therefore seen as a bottom-up activity.

For project C, they decided to collect potentially useful data immediately after the definition of fundamental goals using a one-day workshop. The EF elements like processes, tasks, and product structure were only defined when demand for that occurred. This characterizes an evolutionary approach and is seen by the authors as a bottom-up approach. The main reasons to follow this approach were:

(1) "The immature practices needed to be improved rather quickly, but they did not require highly sophisticated analysis techniques or experience structure documents.
(2) Structures would not be stable anyway.
(3) People were the bottleneck. Effort needed to be concentrated on content first." [5]

The choice between a top-down or bottom-up approach was further influenced by the opinion that stable and mature structures are needed for a top-down approach.

The experiences to date which were of most interest to this project were:

(1) Pros & cons of a top-down approach: The definition of the EF elements in the top-down approach makes it easier for the EF participants to recognize the existence of the EF but provides less concrete early benefits for them. The approach can not be performed without a close connection between the EF and the processes that are in place.

(2) Pros & cons of a bottom-up approach: It may enable a swift realization of the EF results. Results are visible in a short time, but this effect cannot be planned and it is often hard to prove the usefulness beforehand, making visibility of the EF benefits more difficult.

(3) There are many sources of reusable experiences and measurement is just one of them, e.g., intermediate products (like a QA plan) are often seen to be more useful for reuse than concrete experience packages, even when their impact has not been analyzed.

(4) There were no problems in handling and structuring the data. Collecting data and qualitative experiences were the bottleneck.

The EF in the PERFECT project

The PERFECT project is an ESPRIT III project funded by the Central European Commission (CEC) and started in the early 90's. Organizations like Daimler Benz, Siemens, Q-Labs / Ericsson, and the University of Kaiserslautern / Fraunhofer IESE came together with the aim to find a more detailed and tailored approach for the introduction of an EF into organizations. The benefits seen for the approach used include explicit goal setting, focus on products, establishment of a separate organization driving the improvement program, and the tailoring of the activities to specific needs. This is a realization of the principles stated in the EF concept [2].

2. Establishing the EF Goals and Methods

The following points were seen as important in establishing the strategy that would be adopted in the organization.

- Arguments exist that the EF assumes a stable environment, but that this is not suitable for all companies. Their environments may be too dynamic because of short technology cycles. Some organizations argue that stable structures might hinder progress and innovation.
- The time aspect is a critical point. The time frame for first results seems long when following the top-down approach. This can cause problems maintaining participant motivation and management commitment.
- The present EF / QIP approach remains a general, abstract framework, which lacks explicit implementation guidelines and detailed experience reports which are needed in industry. The data that is available at the moment is either experimental or based on a long-term application.
- The EF originated from a scientific and government environment at SEL-NASA and

proved suitable after long-term application. Are the results transferable to software companies in general?

- The bottom-up approach trialled at Daimler-Benz seemed to work as did the top-down approach. There is no detailed data for a comparison of the results of the two approaches. The bottom-up approach brought earlier results. Is a bottom-up approach the better alternative when preliminary processes and an understanding of the environment already exist?
- The EF / QIP approach requires a high degree of experimentation to evaluate techniques. Some companies, especially large ones with R&D departments, have the resources to do that but is it feasible in smaller companies? Often improvement decisions and technology adoptions have to be made much faster than is possible by using pilot projects.
- It is not completely clear whether improvements achieved related to things like reuse and productivity, have their root cause in the introduction of EF concepts or in the successful application of technology. Would the switch to promising techniques such as OO without the introduction of EF have had the same effect?

Setting the project goals

Based on this analysis it was decided that: "The project aim is to develop tools and techniques to improve the speed and quality of software development and to enhance the transfer of process knowledge between projects and project groups." In the organization there were six improvement initiatives present: (1) process tailoring, (2) CMM and ISO 900x assessments, (3) personnel skill improvement, (4) company improvement strategy, (5) self motivated tool development and tool integration (innovative spirit), and (6) the measurement program.

Thus several improvement activities were already present and action plans defined from the results. What the organization needed was a framework to support and focus the related actions. Process tailoring and definition existed and were already applied in parts of the department. Further action was needed to spread them out across the whole department and reuse experience gained during the initial implementations. It was not the main goal to achieve a state such as CMM level-5, which was seen as a hinderence to the company philosophy which was to establish an environment which is reliable and repeatable but not an overly defined one. In the organization the developers initiate a great part of the improvement activities. They identify problems and possible solutions, take ownership and develop solutions in the form of tools or work instructions. This was to be supported and recognized. The present personnel skill improvement activities were to be supported as well as the team spirit and the overall interaction / communication. It was viewed that stable and fixed structures tend to hinder that. The company strategy and goals had been broken down into improvement activities at the project and development level. The project needed to focus and refine these (GQM). The existing measurement program was showing promising results and indicating new improvement items.

It was determined that a bottom-up approach could build on current measurement and initially defined processes and could immediately deliver data associated with known improvement issues. It would also give incentive to the desired tool development. Next we set out to determine whether environmental conditions would also support a bottom-up approach. The situation in each of the project teams is significantly different with respect to techniques and tools deployed. There was an identified need to identify best practices, to document experiences with them, and to support the transfer of knowledge between project teams. It was obvious that the concept of the experience base could help. The information access environment was focused on network technology. Every project team had an internal / external homepage to spread information, they had a project server with related documents, a central mail and document repository existed to get information around and to document daily experiences. Documents templates give the information an identical structure to improve readability and to ensure consistency of data. The mailing and posting repository (Microsoft Outlook) had proved its usefulness in recent years by giving a basis for discussions and to disseminate information. Motivated by the general improvement spirit in the organization, the usage frequency of this repository was fairly high. It was possible to consider using this already-existing, documented experience for an Experience Base (EB). But what was still needed was an effective access technique for the information stored, e.g., a search engine.

Both the normal daily work and solution seeking resulted in high interaction between department members. People were identified as having special knowledge regarding different development fields and the general attitude was to provide others with this knowledge when needed. From unstructured interviews with team members it was identified that it would be useful to package the experiences (daily work knowledge). Initial examinations of the amount of already documented knowledge in reports and mail archives showed that a basic knowledge & experience base already existed on the Intranet but was not yet efficiently usable. Because of the lean hierarchy in the department,

self-motivated improvement activities and the integration of developer opinions was encouraged and simplified. This leads to an environment that is driven dynamically by the team members.

To summarize, information structures were well developed and used in the daily work. This includes the use of network technology as well as the interaction between department members. Highly motivated personnel drive improvement via new techniques, knowledge, and tools. A special focus existed to simplify work tasks through tool support. The daily work and problem solving was strongly based on personnel interaction and access to knowledge bases (documentation, mail lists, etc.). The goal therefore had to be to package personnel experience and best practices and provide an effective framework for access and integration.

From these findings we were convinced that the organization should establish an improvement environment based on the EF concept, but that the appropriate approach was bottom-up.

We defined the EF concept for the organization based on five steps:
Step 1 ***collect*** experience and knowledge,
Step 2 ***publish*** the experience documents and provides an access framework,
Step 3 ***integrate*** experience in an environment were it is needed,
Step 4 ***analyze*** how the experience repository is used, and
Step 5 ***extend*** the structures of the improvement environment when the need occurs.

What is different to the classic EF approach & concept?

The main difference to the EF/QIP concept described in [2] and the concept we describe is in the overall philosophy. First we favored a bottom-up approach starting with providing useable experience from the beginning rather than spending time defining processes and structures for a top-down approach. Moreover, our approach places knowledge management and integration in the center to serve as a driving force for continuous improvement. This is quite different to the EF, which uses the QIP [2] and the GQM [8] as driving forces.

One main concept of the classical Experience Factory is experience generation and explicit experimentation with new technologies to evaluate them and to measure their impact on product and process characteristics. Our approach goes away from explicit experience generation, and focuses on gathering existing experience and supplementing it as it grows using access technology. Furthermore it is not based on the principle of gathering experience from experimentation. Rather the approach uses experience gained with software engineering techniques and new software technologies in the daily work rather than explicitly experimenting with new things. Experience transfer supports the growth of the experience inherent in the environment.

Another difference is that our approach describes how to start the implementation of the first cycle (gathering of existing experience). Our approach allows both the improvement structures and the development environment mature over time. As in the EF framework our experience management environment (EME) supports the documentation and storage of every-day experience. Further more both approaches give structure to establish a continuous improvement environment. The EME is seen to be more evolutionary and able to be adjusted to special needs. The EF gives a predefined structure to be established and therefore changes the existing way to do things.

Requirements for application

Due to the fact that this approach was motivated by environmental characteristics, there exist certain requirements for the application. If another organization intended to apply our approach it should check the following characteristics, which we see as minimal requirements.

- A highly used and developed network environment has to be present and integrated in the daily processes.
- Information repository structures need to be present in the environment, i.e. an Intranet structure using mail archives, project servers, document servers, etc. is needed.
- At least initial processes have to exist, which define when certain information has to be documented, e.g., meeting notes.
- For the documentation style, corporate templates should exist, which give information a common structure.
- Activities have to exist which serve to identify improvement needs outside the knowledge management focus, e.g. CMM assessment.
- There has to be a conviction that there exists a high amount of already documented experience and knowledge in the environment. The document could emphasize things like process documentation, reports, mail archives, web pages, etc.
- The staff have to be self-motivated to search for experience or knowledge.
- The staff have to be self-motivated to experiment with new technologies for the improvement of products and their skills. The

company philosophy should support this by encouraging the staff to do so, e.g., planing time for that and recognizing those activities and the results.

- The organization must have an attitude to let the developers drive changes influenced by strategy and improvement goals. What this also assumes is that there exists a company thinking rather than an individual focus.
- An organization needs resources to establish the concept framework and to maintain it while it matures and grows.
- Initial process and project environment definitions should be available, which build a context for experiences and which can serve as success story examples. At least one project environment should exist, which has documented experience with the introduction of a defined development process.
- The project management staff should be open to constructive suggestion concerning improvements to their development processes.

3. The organizational solution

The project was initiated by a senior manager with a reputation as a successful champion. Staff were involved in all aspects of the initial concept design and subsequent implementation. A project manager from the organization was assigned the task of overseeing the experience factory project. Other staff were involved via seminars, individual consultations and an experience factory website established as a result of requests from the first staff seminar. Thus it was with confidence that we embarked on the technical design and implementation of the experience factory in the organization.

Since we were convinced that the environment already contained a significant amount of documented experience (the mail archive contained around 8500 documents six months after commencement), we began by finding an appropriate technology to gather this experience and make it searchable. Using the tool we selected (Microsoft Site Server 3.0) we also had a framework to make the gathered experience accessible via a web site. Therefore we created a separate web page providing the interface to the indexes. This also provided the integration step into their daily work. When someone wanted to find existing experience about a task or general information from the environment they could now do that using the web page.

Evaluation of indexing tools

The first step in the evaluation was to define the requirements for an appropriate indexing tool. These were separated into two kinds of requirements. We defined requirements for a surface-evaluation, i.e. an initial evaluation to check basic functionality, and when a tool passed these we evaluated it against further interface and behavior related requirements. In Table 1 they are marked as RSx (surface) and RDx (detailed) requirements:

Table 1. Requirements

Requirement	Description
RS1	*Price*: the price of the tool shall be reasonable, preferably freeware.
RS2	*File types*: the tool shall be able to create an index of common files including Microsoft Office documents, HTML, PDF, MS exchange files.
RS3	*Interface type*: the tool shall provide a web-based interface to apply queries on the document index to find appropriate information. It shall at least be possible to redirect the query input and output from a web site to the tool and vice versa.
RS4	*Scalability*: the tool shall be scalable, i.e. the amount of indexed documents and users should not be limited.
RS5	*Gathering*: the tool shall be able to gather documents over a Microsoft NT network. Tools running on a Unix machine but able to access NT would also fulfill this requirement.
RD1	*Performance*: the tool shall be reasonable fast, e.g., search queries shall be answered in less than a minute, re-indexing shall be possible over a weekend.
RD2	*Maintenance*: the tool shall provide a mechanism to automatically update the search base (scheduled builds).
RD3	*Interface style*: it shall be possible to provide the user with a short description of query matching documents and to modify the style of the interface.
RD4	*Access rights*: The tool shall be able to give a user only access to those files to which he has access over the NT network.

Based on these results we decided after three weeks of tool evaluation to use Microsoft Site Server 3.0 (MSS) as the tool to gather and publish our environment, experience & knowledge documents, to build the base for our experience management environment (EME). The network environment (Intranet) was a Microsoft Windows NT network. Clients were running Windows 98, Windows NT, or Windows NT Server. Furthermore a couple of servers running Unix are connected and their file system can be accessed over the NT network from other non-Unix machines. The document sources are summarized in Table 2.

Table 2. Document Sources

Document source	File types	Information type
Mail Exchange Server	Mail format (exch)	Folders for past and current project information, technology discussions, reuse items, etc.
Project and department web server	HTML, Microsoft Office documents (DOC, XLS, PPT), Adobe Acrobat PDF, database files (SQL, Access)	Project documents ranging from code to process descriptions, general department information like administration tasks
Local workstations	Microsoft Office documents (DOC, XLS, PPT), Adobe Acrobat PDF, HTML, plain text (TXT)	Documents gathered for own information purpose, document drafts

4. Analysis of Usage

The analysis step consisted of preliminary analysis of the access log data to the web. The tool provided us with the functionality to create usage reports. In addition we conducted a survey on the benefits the people observed while using it. In three months we were able to implement the structures for the four steps 'collect', 'publish', 'integrate', and 'analyze'. We were able to identify items for extension activities (step 'extend') from these results and from user feedback, which helped to focus on the future.

The growth of our document search repository over time was influenced by three factors:

- including more document sources in a specific catalog,
- including more types of documents in a catalog definition, and
- the growth of experience & knowledge documents in the environment over time.

The number of documents changed with every build cycle for the catalogs. Numbers of gathered documents in the search repository have been documented and are shown in Figure 1, together with the factor which mainly influenced the growth. Here we see a significant growth in documents available after a relatively short period of time.

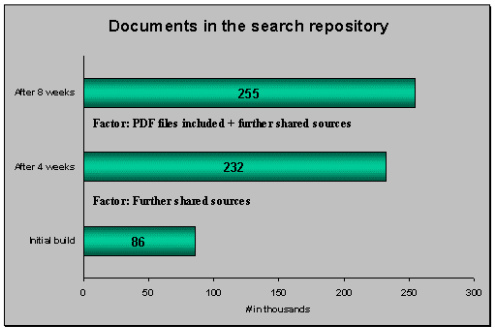

Figure 1. Growth of Documents in the Search Repository

Figure 2 shows the growth of the mail repository which results largely from daily work. The figure shows that in the last four weeks of the project the growth in the mail repository was 1,800 new documents. This is not to say that every added document is indeed useful as a reusable experience, but it indicates that daily work items were documented and shared.

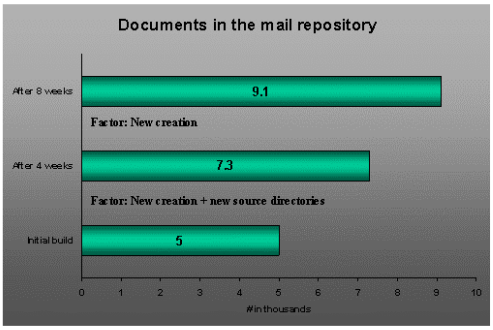

Figure 2 Growth of documents in the mail repository

In figure 3 we show the use of the repository over a seven week period. This figure shows that, in the early stages, people became more and more aware of the repository and more people tested the repository with their personal information needs (peak in the third bar). After that the usage frequency was lower, more stable and continuous,

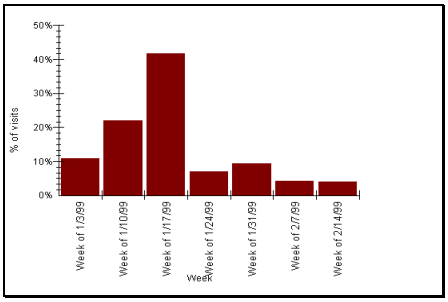

Figure 3. Usage of the Search Repository per week

indicating possible acceptance. In this figure, a visit is defined as a series of consecutive requests from a user to an Internet site and a request is a successful connection to an Internet site, i.e. retrieving contents. The graph shows the distribution of the number of different users visiting the web site as a percent of total visits over the period.

Figure 4 shows the average number of queries entered per visit per week. The number is low at the beginning. People were testing the repository with an average of one query, presumably to see the behavior and the functionality. Later the people seem to search more seriously for information. Further information provided to the department about the intent and use of the search repository probably caused the high increase at the end. The

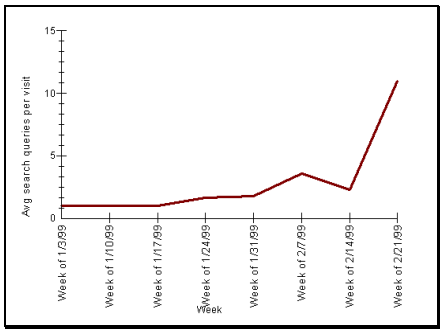

Figure 4. Average search queries per visit

combination of figures 3 and 4 is interesting. It shows that number of visits seems to be stabilizing but that the number of queries per visit is increasing. This demonstrates a relatively efficient usage pattern.

The more popular search queries entered during the last 8 weeks of the project were basically a binary classification of technology issues and process issues. The technology issues include ActiveX and XML. The process issues were classified as "process" in general and "estimation process". The data indicates a large diversity of information needs in the organizational environment. The repository was able to give back possibly useful documents for most queries. However we do not have any information to indicate whether the returned documents were useful or not.

Some queries did not return documents since the repository contained insufficient documents, for example new technologies like the XML language. As with the usage report, we need to be careful with the query data because it is only initial data from a short period of use.

The reports, although only initial, provide some preliminary indications.

- The acceptance, i.e. usage of the repository was promising.
- The usage frequency indicates a degree of integration into the daily information search activities.
- The information which was needed in the department covers a very wide range of areas.
- Process and new technology information seems to be of special interest.
- Informing people about the presence and the usefulness of the concept is important.

These were the initial conclusions from the limited data available. Surveying the repository users then extended these.

Survey about usage benefits

To get direct user feedback we decided to conduct an informal survey of staff impressions while using the search engine, ideas the users had, and the benefits realized through being able to search the local environment documents. Overall the acceptance and judgement of the product was good. The feedback ranged from ideas for extension, descriptions of how people used the repository, to first impressions. The following points capture the most common critical aspects, benefits, and extension ideas gained from the survey.

We found that the people who had been working in the department for a long time knew where information could be found without using the repository (e.g. document templates or whom to ask to get information). The opportunity for this will reduce as the department grows. We would then predict that the repository could play a stronger role in information transfer.

The benefits that were noted included comment that the search web site is a good address for new employees who are not familiar with the work environment. People also reported that they found documents and information that had been lost. The average time saved through this was estimated to be in a range of 1 to 4 hours. The search engine also reportedly breaks down information barriers between projects and environments (sharing experience & knowledge). It was seen as a good thing to first search for local information and experience before proceeding.

6. Conclusions

At the end of our implementation of cycle #1 we assessed which of our initial expectations for the defined approach were met. Earlier we described our expectations, which we now examine. The time our experience management environment (EME) was usable was 8 weeks and hence the underlying data has to be viewed carefully and further trends have to be monitored to prove the findings.

Our experience is that we generally achieved the technical objectives. In this respect the project was successful. We are relatively confident that the experience management environment could help support improvement in this environment. The data that was available at this time was too preliminary to justify strong conclusions about usage of the experience base. The usage patterns indicated a trend towards consistent use and integration into the work cycle. The project proved the viability of the bottom-up approach selected in this organization. Whether this will apply in other organizations clearly depends on many factors. We have outlined what we believe these factors to be. They range from broad organizational and cultural characteristics to technology characteristics. The most important evidence, we believe, is the clear establishment of a substantial experience base in an organizational setting in a short time period, which showed indications of successful deployment.

So what went wrong? Surprisingly, given the positive comments by the users, the system was decommissioned shortly after the completion of the project. A major contributor to this was the lack of ongoing management commitment to the project. While a senior manager was the initial champion of the project, its implementation was assigned to a busy project leader. In retrospect greater emphasis should have been placed on ensuring that the project champion maintained a more visible presence with respect to the experience factory project. A second issue was the lack of identification of clear goals and payback criteria for the project. It appears that, although technology can support this type of experience base development, a top down GQM-based methodology has the characteristics that are more likely to ensure longer-term success. The third observation was that the close physical proximity of the development teams and the relatively small number of personnel worked against the need for a more formal repository-based experience factory. The metrics success factors documented by Jeffery and Berry in [9] might provide an indicator of factors relevant to EF success as well. For example they list senior management commitment, realistic assessment of payback, clear responsibilities, determination of required granularity among many others. The issue of physical proximity has been observed by the authors in the context of electronic conferencing as a major implementation issue.

7. References

[1] Birk A. and Tutz C., "Knowledge Management of Software Engineering Lessons Learned", Fraunhofer IESE-Report No. 002.98/E, July, 1998

[2] Basili V., Caldiera G. and Rombach D., "The Experience Factory", in Marciniak J.J. ed., *Encyclopedia of Software Engineering*, John Wiley & Sons, 1994.

[3] PERFECT homepage, ESPRIT III project sponsored by the CEC, http://www.iese.fhg.de/Services/Projects/Public-Projects/Perfect/perfect.html

[4] Basili V., Caldiera G., McGarry F., Pajerski R., Page G. and Waligora S., "The Software Engineering Laboratory – An Operational Software Experience Factory", *Proc. of the International Conference on Software Engineering*, Melb, May, 1992, pp.370-381.

[5] Houdek F., Schneider K. and Wieser E., "Establishing Experience Factories at Daimler Benz: An Experience Report, *IEEE*, 1998, pp. 443-447.

[6] Basilli V. and Caldiera G., "Improve Software Quality by reusing Knowledge and Experience", *Sloan Management Review*, 37(1), 1995, pp. 55-64

[7] Basili V. and McGarry F. "The Experience Factory: How to Build and Run One", 17th International Conference on Software Engineering, Seattle USA, 1995

[8] Feldmann R. and Vorwieger S., "The web-based Interface to the SFB 501 Experience Base", SFB 501 Bericht 01/91, University of Kaiserslautern, 1991.

[9] Jeffery, R. and Berry, M. "A Framework for Evaluation and Prediction of Metrics Program Success", in Applying Software Metrics, edited by Paul Oman and Shari Lawrence Pfleeger, IEEE Computer Society, 1997, pp. 266-277.